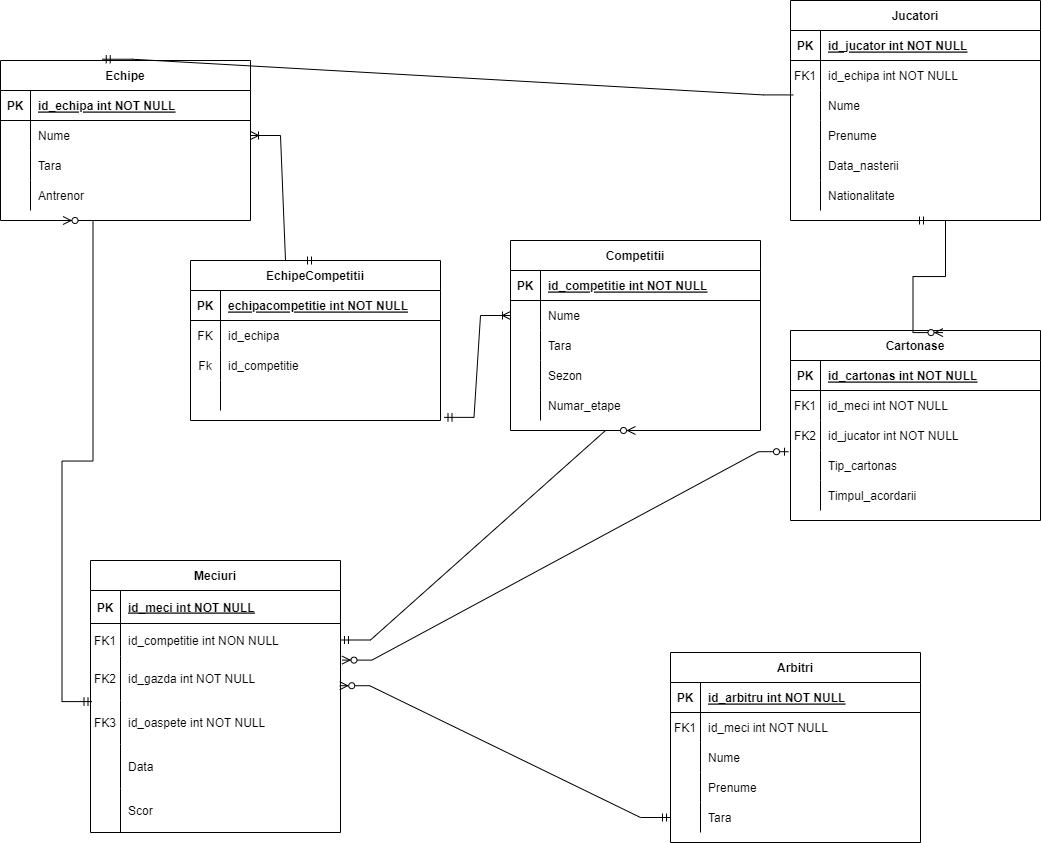
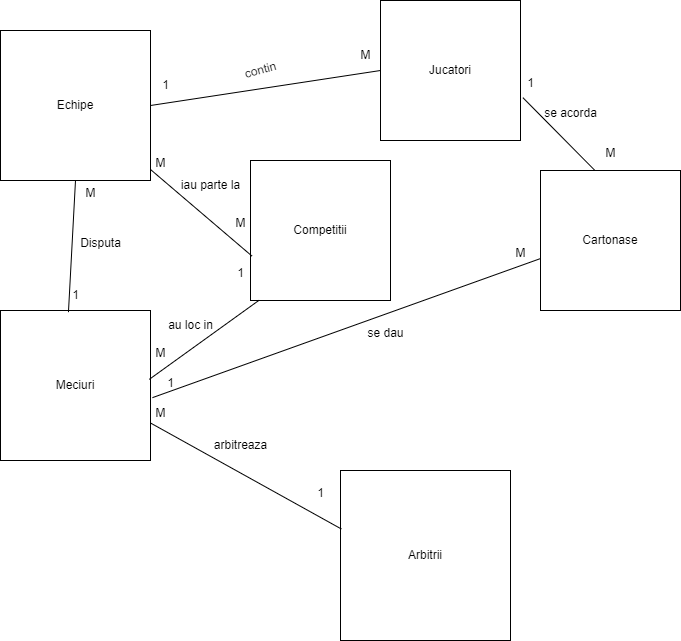
**Proiect  
   
 SISTEME DE GESTIUNE A   
 BAZELOR DE DATE** FURDUI VLAD RAREȘ  
   
 - GRUPA 241-  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**1. PREZENTAREA PROIECTULUI** Acest proiect are ca scop organizarea unui sezon fotbalistic ce implică jucători, echipe, competiții, arbitrii, cartonașe și meciuri.  
  
 Un arbitru poate arbitra mai multe meciuri, poate da mai multe cartonașe jucătorilor.  
  
 Fiecare jucător evoluează la o echipă care, de asemenea, poate să fie angrenată în mai multe competiții, fie la nivel de țară sau continent.  
   
 Meciurile se desfășoară în cadrul unei competiții, la care părtașe sunt două echipe, una gazdă și una oaspete.   
 **2. Diagrama ERD**



**3. Diagrama conceptuală**



**4. Creare tabele**

CREATE TABLE Echipe (

ID\_Echipa NUMBER PRIMARY KEY,

Nume VARCHAR2(100),

Tara VARCHAR2(100),

Antrenor VARCHAR2(100)

);

CREATE TABLE Jucatori (

ID\_Jucator NUMBER PRIMARY KEY,

Nume VARCHAR2(100),

Prenume VARCHAR2(100),

Data\_nasterii DATE,

Nationalitate VARCHAR2(100),

ID\_Echipa NUMBER,

CONSTRAINT fk\_jucatori\_echipa FOREIGN KEY (ID\_Echipa) REFERENCES Echipe (ID\_Echipa)

);

CREATE TABLE Competitii (

ID\_Competitie NUMBER PRIMARY KEY,

Nume VARCHAR2(100),

Tara VARCHAR2(100),

Sezon VARCHAR2(100),

Numar\_etape NUMBER

);

CREATE TABLE Echipe\_Competitii (

ID\_EchipaCompetitie NUMBER,

ID\_Echipa NUMBER,

ID\_Competitie NUMBER,

CONSTRAINT pk\_echipe\_competitii PRIMARY KEY (ID\_EchipaCompetitie),

CONSTRAINT fk\_echipe\_competitii\_echipa FOREIGN KEY (ID\_Echipa) REFERENCES Echipe (ID\_Echipa),

CONSTRAINT fk\_echipe\_competitii\_comp FOREIGN KEY (ID\_Competitie) REFERENCES Competitii (ID\_Competitie)

);

CREATE TABLE Meciuri (

ID\_Meci NUMBER PRIMARY KEY,

ID\_Gazda NUMBER,

ID\_Oaspete NUMBER,

ID\_Competitie NUMBER,

Data DATE,

Scor VARCHAR2(100),

CONSTRAINT fk\_meciuri\_echipa\_gazda FOREIGN KEY (ID\_Gazda) REFERENCES Echipe (ID\_Echipa),

CONSTRAINT fk\_meciuri\_echipa\_oaspete FOREIGN KEY (ID\_Oaspete) REFERENCES Echipe (ID\_Echipa),

CONSTRAINT fk\_meciuri\_competitie FOREIGN KEY (ID\_Competitie) REFERENCES Competitii (ID\_Competitie)

);

CREATE TABLE Arbitri (

ID\_Arbitru NUMBER PRIMARY KEY,

Nume VARCHAR2(100),

Prenume VARCHAR2(100),

Tara VARCHAR2(100),

ID\_Meci NUMBER,

CONSTRAINT fk\_arbitri\_meciuri FOREIGN KEY (ID\_Meci) REFERENCES Meciuri (ID\_Meci)

);

CREATE TABLE Cartonase (

ID\_Cartonas NUMBER PRIMARY KEY,

ID\_Meci NUMBER,

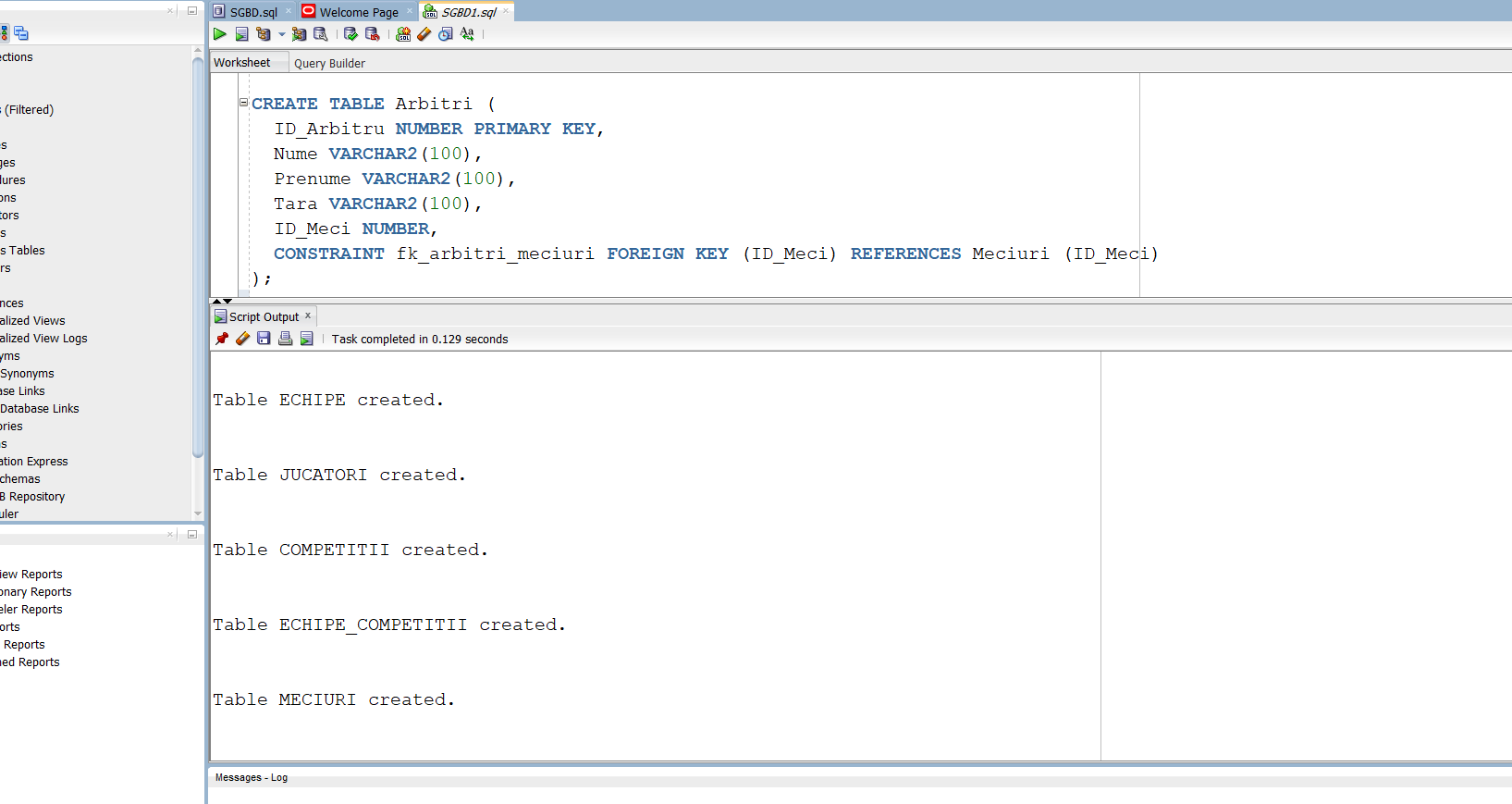
ID\_Jucator NUMBER,

Tip\_cartonas VARCHAR2(100),

Timpul\_acordarii NUMBER,

CONSTRAINT fk\_cartonase\_meci FOREIGN KEY (ID\_Meci) REFERENCES Meciuri (ID\_Meci),

CONSTRAINT fk\_cartonase\_jucator FOREIGN KEY (ID\_Jucator) REFERENCES Jucatori (ID\_Jucator)

);  
  
  
  
  
  
**5. Inserare date**  
INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (1, 'Steaua Bucuresti', 'Romania', 'Daniel Oprita');

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (2, 'Dinamo Bucuresti', 'Romania', 'Ovidiu Burca');

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (3, 'Liverpool', 'Anglia', 'Jurgen Klopp');

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (4, 'Manchester United', 'Anglia', 'Ole Gunnar Solskjaer');

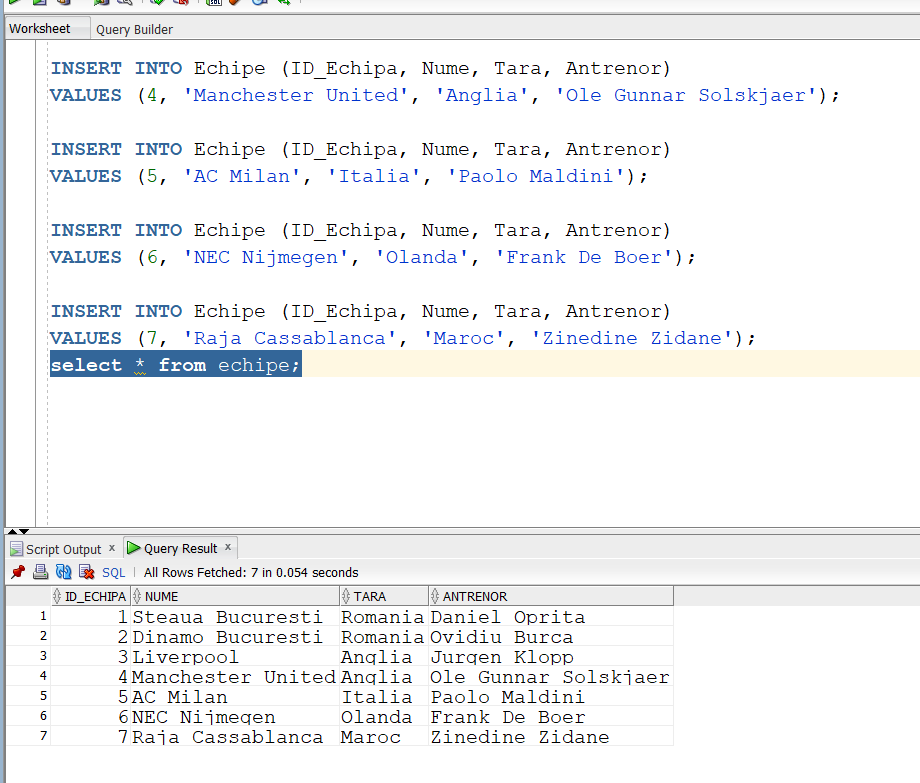
INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (5, 'AC Milan', 'Italia', 'Paolo Maldini');

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (6, 'NEC Nijmegen', 'Olanda', 'Frank De Boer');

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (7, 'Raja Cassablanca', 'Maroc', 'Zinedine Zidane');  
  
INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (1, 'Chipirliu', 'Remus', TO\_DATE('1984-06-15', 'YYYY-MM-DD'), 'Roman', 1);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (2, 'Mihaescu', 'Mihai', TO\_DATE('1990-01-02', 'YYYY-MM-DD'), 'Roman', 1);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (3, 'Bani', 'Ahmed', TO\_DATE('1999-10-10', 'YYYY-MM-DD'), 'Roman', 2);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (4, 'Salah', 'Mo', TO\_DATE('1995-11-11', 'YYYY-MM-DD'), 'Egiptean', 3);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (5, 'Shaw', 'Luke', TO\_DATE('1988-03-03', 'YYYY-MM-DD'), 'Englez', 4);

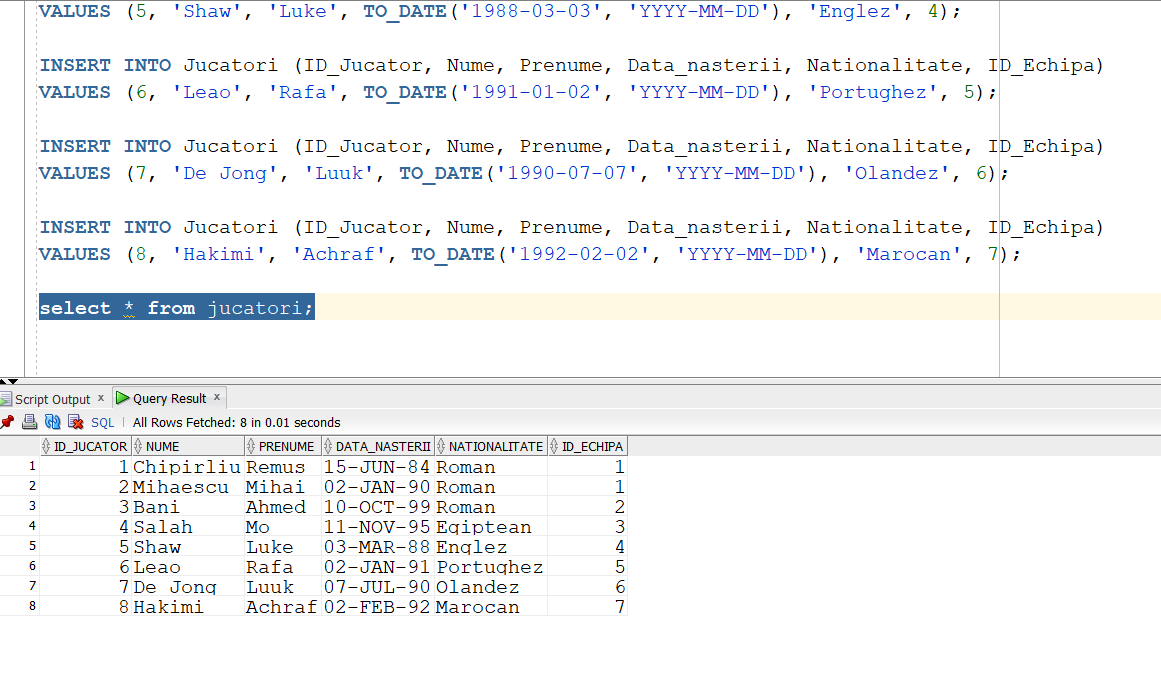
INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (6, 'Leao', 'Rafa', TO\_DATE('1991-01-02', 'YYYY-MM-DD'), 'Portughez', 5);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (7, 'De Jong', 'Luuk', TO\_DATE('1990-07-07', 'YYYY-MM-DD'), 'Olandez', 6);

INSERT INTO Jucatori (ID\_Jucator, Nume, Prenume, Data\_nasterii, Nationalitate, ID\_Echipa)

VALUES (8, 'Hakimi', 'Achraf', TO\_DATE('1992-02-02', 'YYYY-MM-DD'), 'Marocan', 7);  
  
INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (1, 'UEFA Champions League', 'Europa', '2021-2022', 8);

INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (2, 'Premier League', 'Anglia', '2021-2022', 38);

INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (3, 'Serie A', 'Italia', '2021-2022', 30);

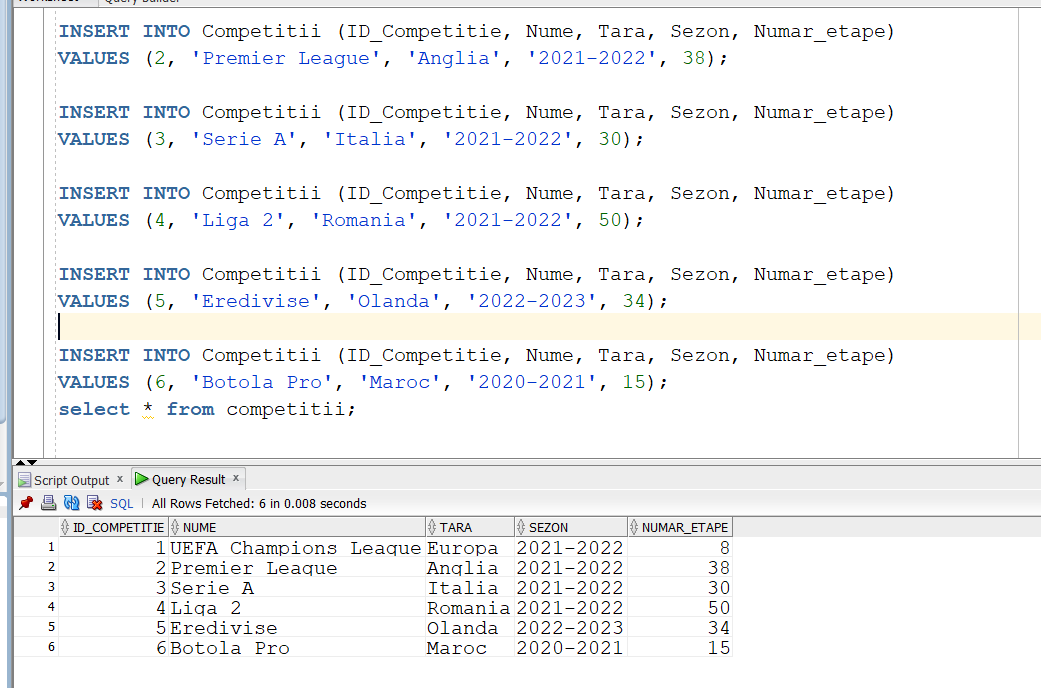
INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (4, 'Liga 2', 'Romania', '2021-2022', 50);

INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (5, 'Eredivise', 'Olanda', '2022-2023', 34);

INSERT INTO Competitii (ID\_Competitie, Nume, Tara, Sezon, Numar\_etape)

VALUES (6, 'Botola Pro', 'Maroc', '2020-2021', 15);  
  
  
  
INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (1, 3, 1);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (2, 4, 1);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (3, 5, 1);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (4, 3, 2);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (5, 4, 2);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (6, 1, 4);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (7, 2, 4);

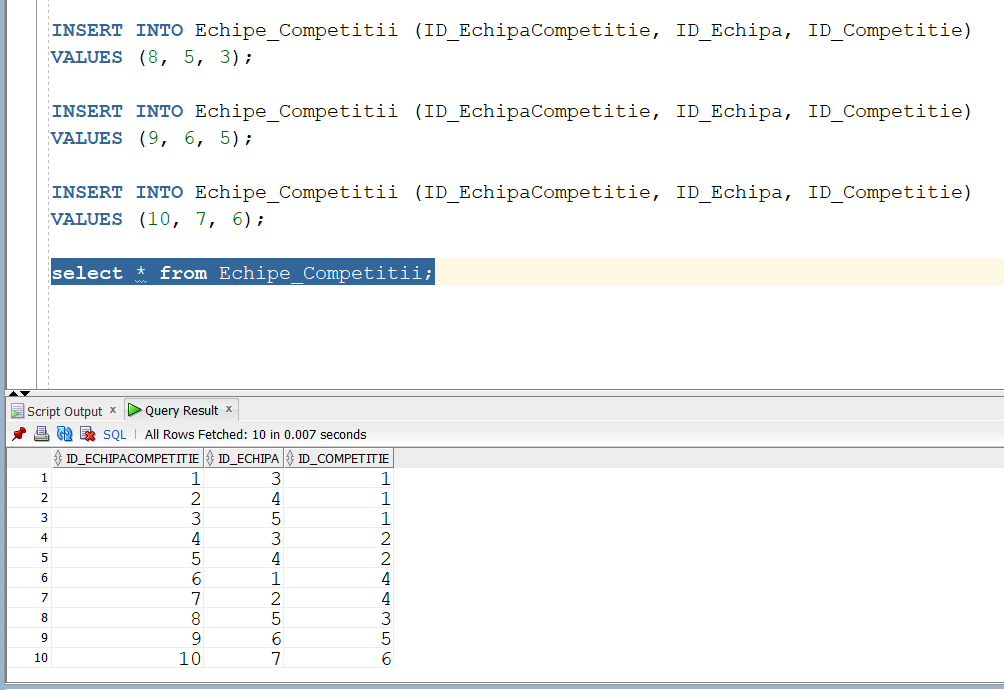
INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (8, 5, 3);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (9, 6, 5);

INSERT INTO Echipe\_Competitii (ID\_EchipaCompetitie, ID\_Echipa, ID\_Competitie)

VALUES (10, 7, 6);  
  
INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (1, 4, 6, 1, TO\_DATE('2022-03-12', 'YYYY-MM-DD'), '4-4');

INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (2, 3, 5, 1, TO\_DATE('2021-11-16', 'YYYY-MM-DD'), '1-0');

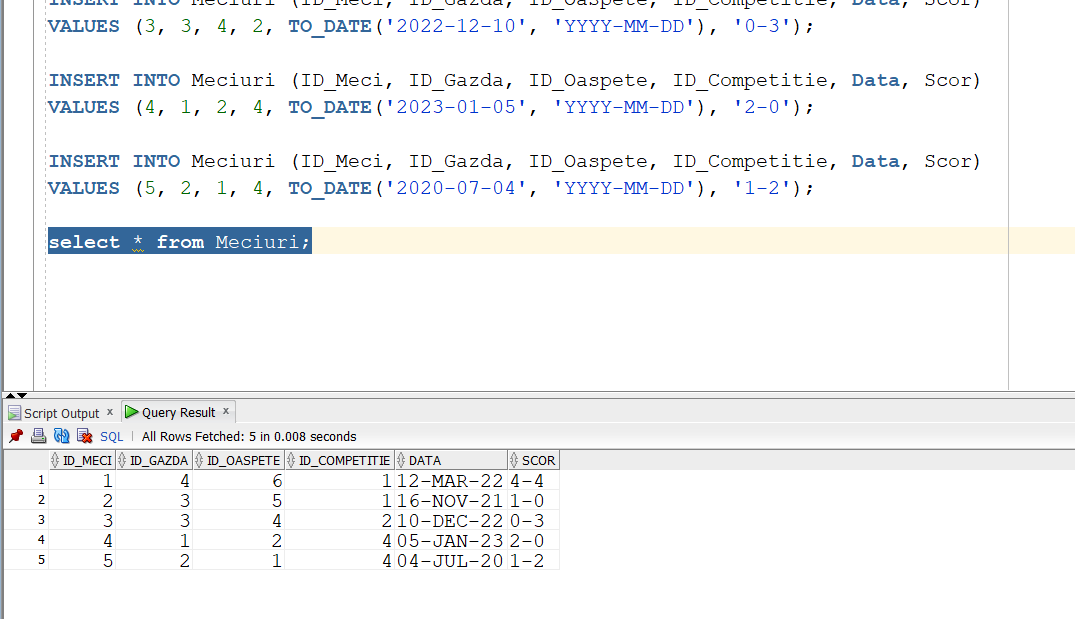
INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (3, 3, 4, 2, TO\_DATE('2022-12-10', 'YYYY-MM-DD'), '0-3');

INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (4, 1, 2, 4, TO\_DATE('2023-01-05', 'YYYY-MM-DD'), '2-0');

INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (5, 2, 1, 4, TO\_DATE('2020-07-04', 'YYYY-MM-DD'), '1-2');  
  
  
  
INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

VALUES (1, 'Coleman', 'Ronnie', 'America', 1);

INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

VALUES (2, 'Waygu', 'Ho', 'Burundi', 2);

INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

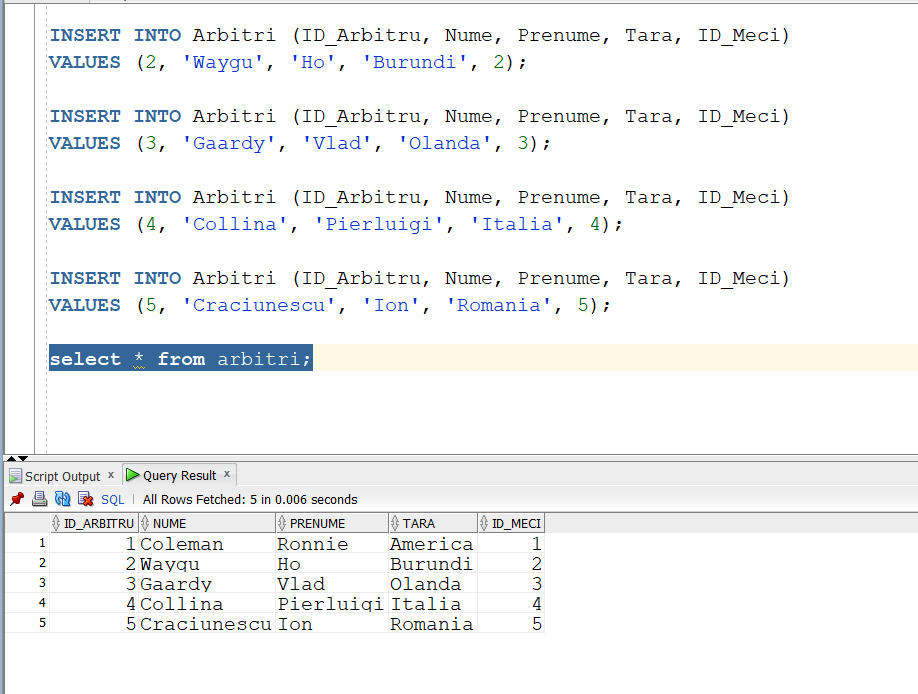
VALUES (3, 'Gaardy', 'Vlad', 'Olanda', 3);

INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

VALUES (4, 'Collina', 'Pierluigi', 'Italia', 4);

INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

VALUES (5, 'Craciunescu', 'Ion', 'Romania', 5);

  
  
INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (1, 1, 1, 'Galben', 25);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (2, 2, 4, 'Rosu', 45);

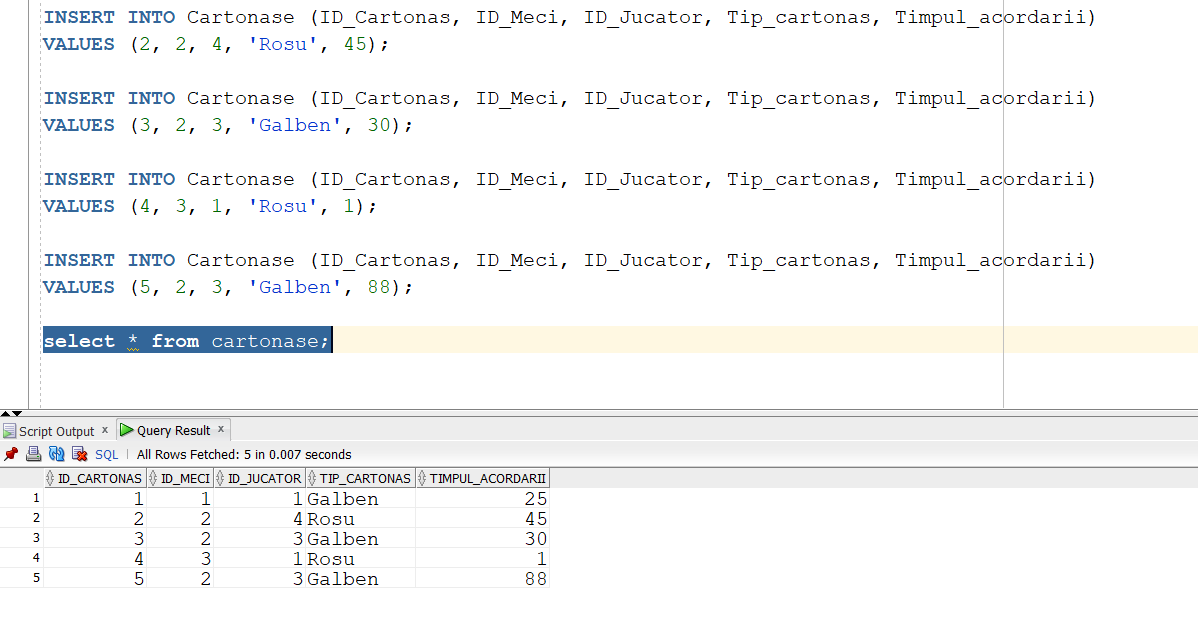
INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (3, 2, 3, 'Galben', 30);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (4, 3, 1, 'Rosu', 1);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (5, 2, 3, 'Galben', 88);  
  
  
  
  
**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.   
  
-6. Sa se memoreze si afiseze, pentru fiecare echipa, toti jucatorii nascuti dupa 1990**

CREATE OR REPLACE PROCEDURE ex\_6 AS

TYPE jucatori IS RECORD (

nume VARCHAR2(100),

prenume VARCHAR2(100)

); --tip de date record pentru a memora numele si prenumele

TYPE jucator IS TABLE OF jucatori INDEX BY PLS\_INTEGER; --tablou indexat pentru a retine numele si prenumele fiecarui jucator

TYPE toti\_jucatorii IS TABLE OF jucator; --tablou imbricat pentru a retine jucatorii pentru fiecare echipa

players toti\_jucatorii := toti\_jucatorii();

nume\_echipa ECHIPE.nume%type;

BEGIN

FOR i IN (SELECT id\_echipa FROM ECHIPE) LOOP --i va fi fiecare id al unei echipe in parte

players.DELETE; --players se va reseta pentru fiecare echipa

FOR player IN (SELECT nume, prenume FROM jucatori WHERE id\_echipa = i.id\_echipa AND data\_nasterii > TO\_DATE('1990-01-01', 'YYYY-MM-DD')) LOOP

players.EXTEND; --aplicarea conditiilor si adaugarea jucatorilor

players(players.COUNT)(players(players.COUNT).COUNT+1) := player;

END LOOP;

SELECT nume

INTO nume\_echipa

FROM ECHIPE e

WHERE i.id\_echipa = e.id\_echipa; --variabila pentru stocarea echipei ca o sa fie afisata

DBMS\_OUTPUT.PUT\_LINE(nume\_echipa);

FOR j IN 1..players.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(players(j)(1).nume); --afisarea numelui si a prenumelui

DBMS\_OUTPUT.PUT\_LINE(players(j)(1).prenume);

END LOOP;

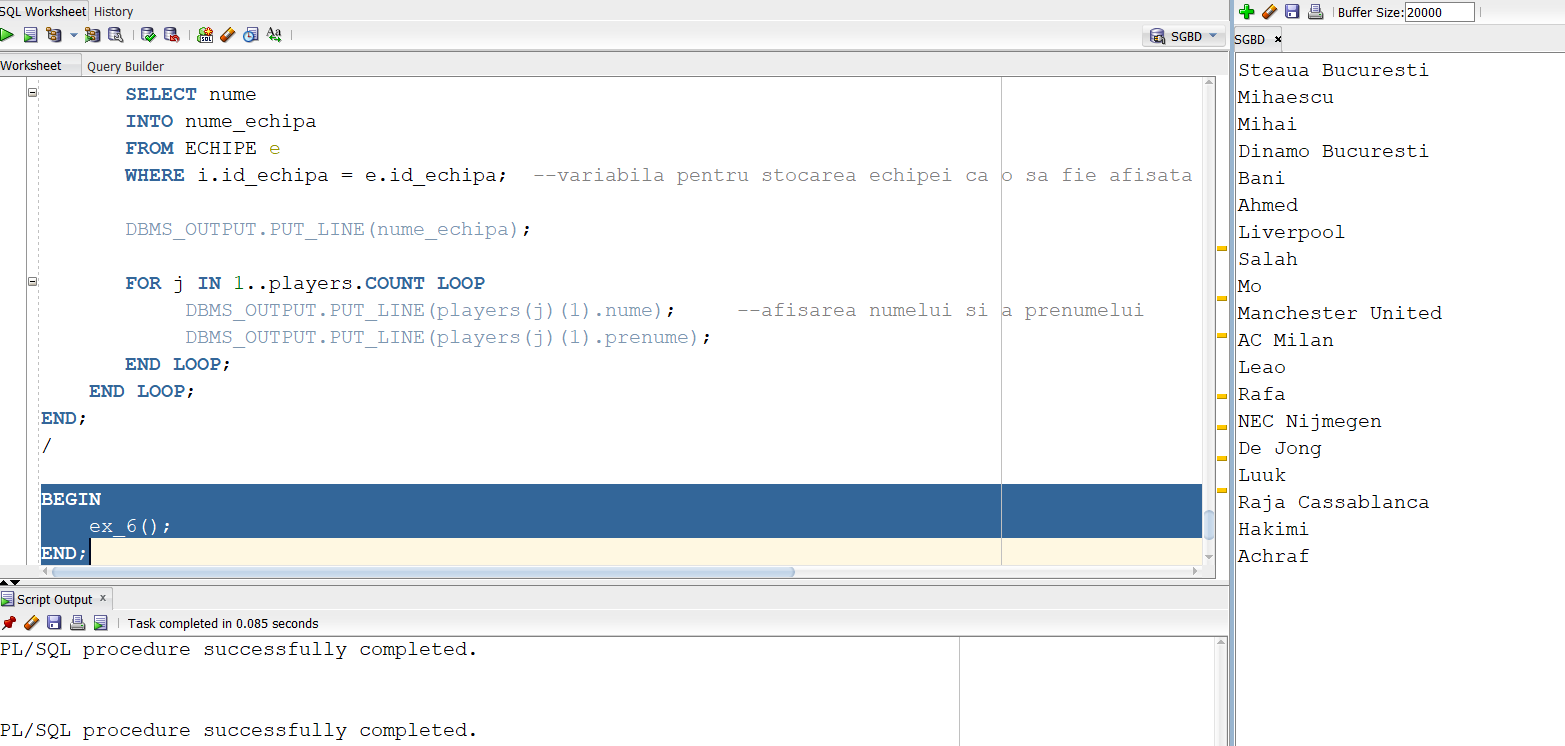
END LOOP;

END;

/

BEGIN

ex\_6();

END; **   
  
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.   
  
--7 Sa se afiseze pentru fiecare echipa numele jucatorilor**

CREATE OR REPLACE PROCEDURE ex\_7 AS

Cursor teams is --cursor care selecteaza id-ul fiecarei echipe

SELECT id\_echipa

FROM ECHIPE;

CURSOR players (echipa integer) is --cursor parametrizat ce selecteaza in baza

--id-ului furnizat numele si prenumele jucatorilor

Select nume, prenume

FROM Jucatori

Where id\_echipa = echipa;

team echipe.id\_echipa%type;

nJuc jucatori.nume%type;

pJuc jucatori.prenume%type;

team\_name echipe.nume%type;

BEGIN

for ech in teams LOOP

team := ech.id\_echipa;

--se itereaza prin toate id\_urile

Select nume into team\_name FROM echipe where id\_echipa = team; --se selecteaza numele echipei

DBMS\_OUTPUT.PUT\_LINE(team\_name);

for var in players(team) LOOP --se itereaza cu parametru id\_ul echipei numele si prenumele fiecarui jucator

nJuc := var.nume;

pJuc := var.prenume;

DBMS\_OUTPUT.PUT\_LINE(nJuc);

DBMS\_OUTPUT.PUT\_LINE(pJuc);

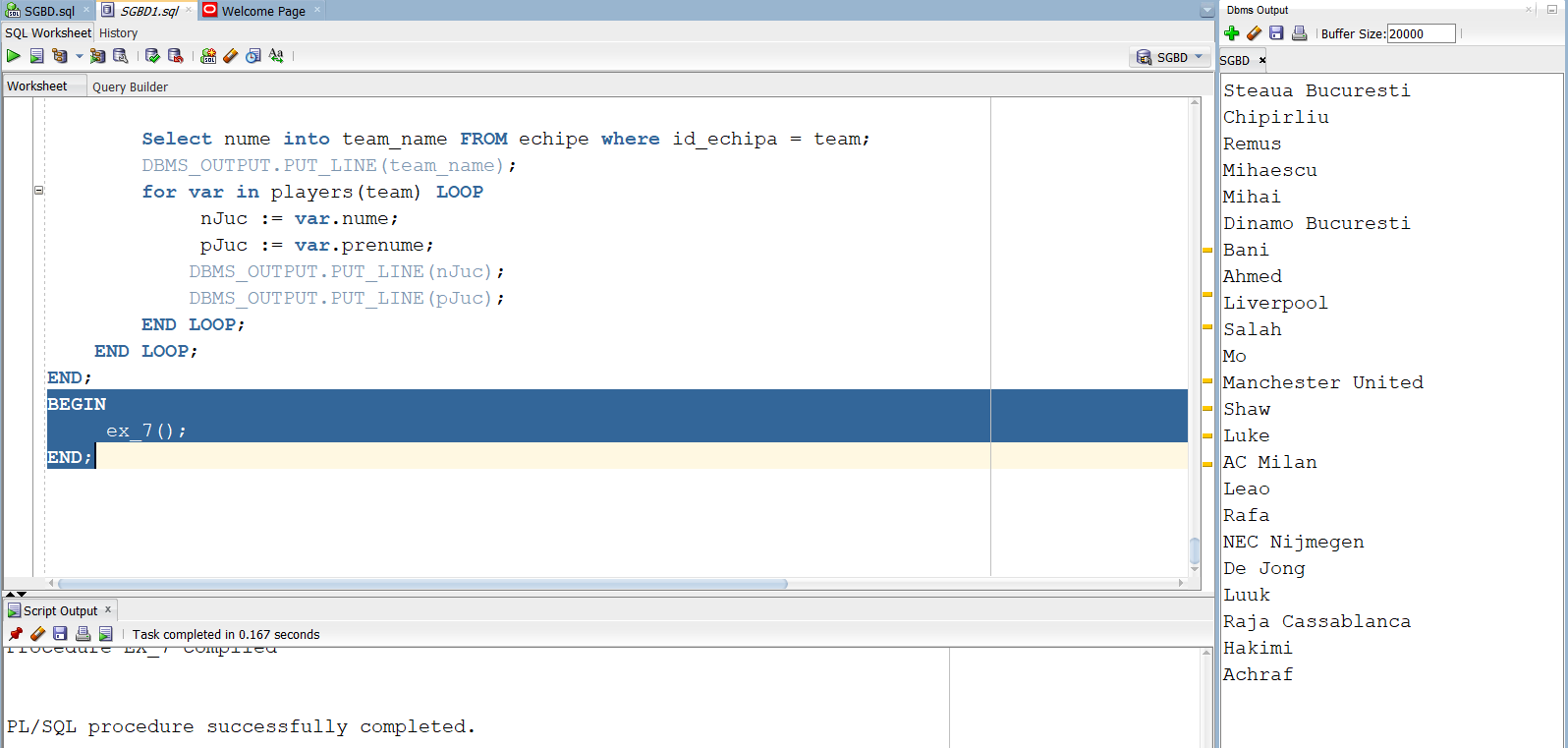
END LOOP;

END LOOP;

END;

BEGIN

ex\_7(); --se apeleaza procedura

END; ****  
  
**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.**  
  
  
**--8 sa se afiseze echipele care au primit un cartonas galben in a doua repriza in timpul unui meci dat**

CREATE OR REPLACE FUNCTION ex\_8(id\_card INTEGER) RETURN VARCHAR2 IS --se va primi prin id card id-ul meciului

exceptie\_rosu EXCEPTION;

exceptie\_prima\_repriza EXCEPTION;

nume echipe.nume%TYPE;

tip\_cartonas cartonase.tip\_cartonas%TYPE;

timpul\_acordarii cartonase.timpul\_acordarii%TYPE;

BEGIN

SELECT c.tip\_cartonas, c.timpul\_acordarii

INTO tip\_cartonas, timpul\_acordarii --se vor selecta tipul si timpul acordarii cartonasului

FROM cartonase c

WHERE c.id\_cartonas = id\_card;

IF tip\_cartonas = 'Rosu' THEN

RAISE exceptie\_rosu;

ELSIF timpul\_acordarii < 45 THEN --se vor verifica cele 2 conditii impuse de cerinta, ulterior intrandu-se in exceptie

--daca nu se indeplinesc

RAISE exceptie\_prima\_repriza;

ELSE

SELECT e.nume

INTO nume

FROM cartonase c, meciuri m, echipe e --se vor utiliza trei tabele

WHERE c.id\_cartonas = id\_card and c.id\_meci = m.id\_meci and m.id\_oaspete = e.id\_echipa ;

RETURN nume; --se va returna numele echipei

END IF;

EXCEPTION

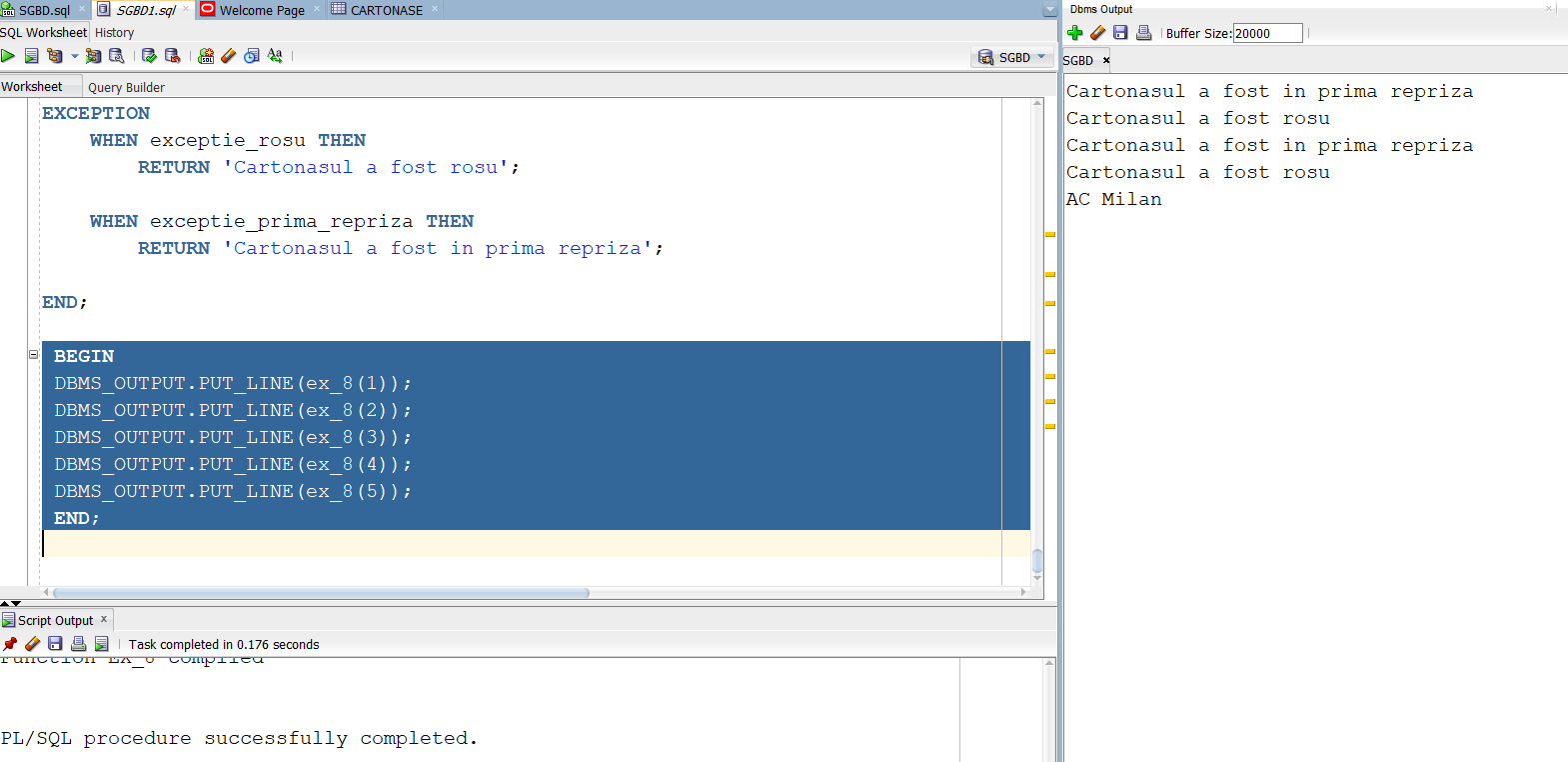
WHEN exceptie\_rosu THEN

RETURN 'Cartonasul a fost rosu'; --se vor returna mesaje aferente exceptiilor

WHEN exceptie\_prima\_repriza THEN

RETURN 'Cartonasul a fost in prima repriza';

END;

**  
  
9 Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.  
  
  
Afișați jucătorul care a obținut roșu de la un arbitru de aceeași naționalitate când echipa sa juca in deplasare.**În prealabil am schimbat toate naționalitățile jucătorilor cu numele țării.  
  
UPDATE jucatori

SET nationalitate = 'Romania'

WHERE id\_jucator = 1;

UPDATE jucatori

SET nationalitate = 'Romania'

WHERE id\_jucator = 2;

UPDATE jucatori

SET nationalitate = 'Egipt'

WHERE id\_jucator = 4;

UPDATE jucatori

SET nationalitate = 'Anglia'

WHERE id\_jucator = 5;

UPDATE jucatori

SET nationalitate = 'Portugalia'

WHERE id\_jucator = 6;

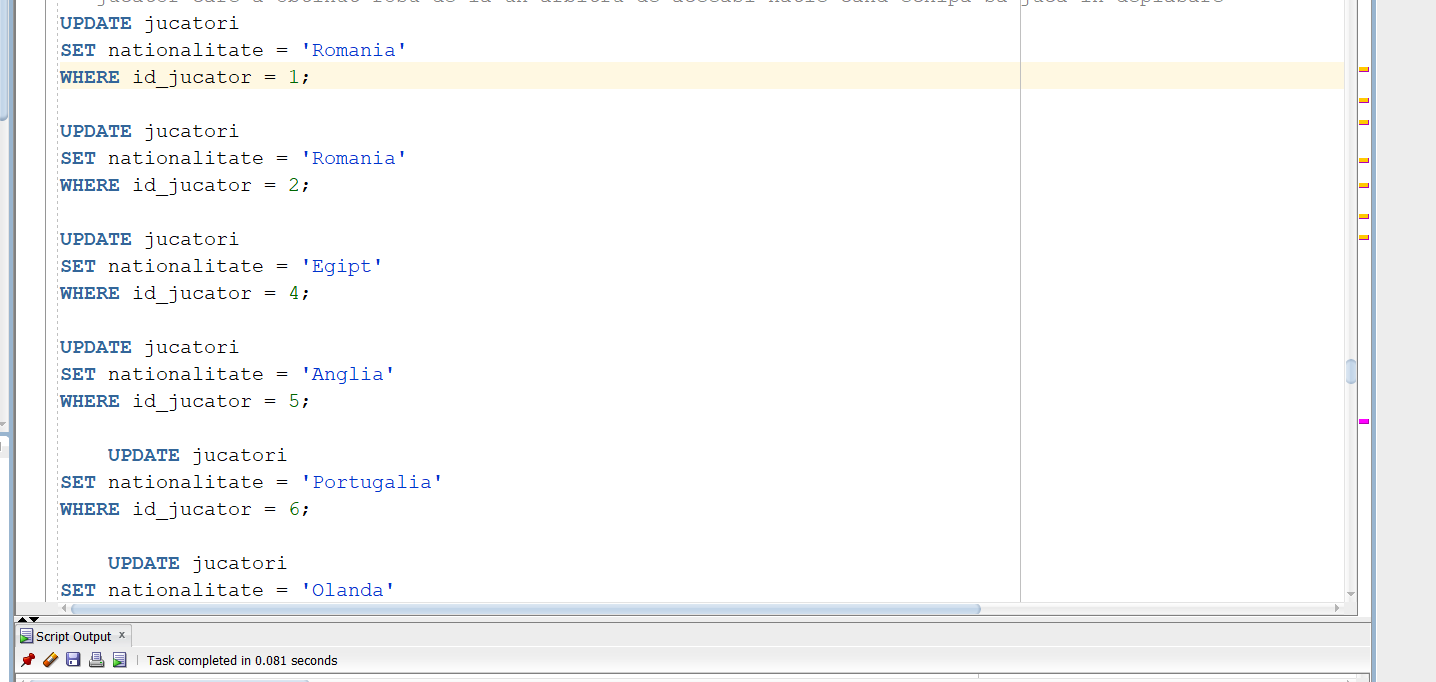
UPDATE jucatori

SET nationalitate = 'Olanda'

WHERE id\_jucator = 7;

UPDATE jucatori

SET nationalitate = 'Maroc'

WHERE id\_jucator = 8;  
  
****  
  
Și am adăugat date în tabel pentru a putea exemplifica toate cazurile posibile ce pot apărea în subprogram   
  
INSERT INTO Meciuri (ID\_Meci, ID\_Gazda, ID\_Oaspete, ID\_Competitie, Data, Scor)

VALUES (6, 2, 1, 4, TO\_DATE('2023-04-04', 'YYYY-MM-DD'), '3-0');

INSERT INTO Arbitri (ID\_Arbitru, Nume, Prenume, Tara, ID\_Meci)

VALUES (6, 'Hategan', 'Ovidiu', 'Romania', 6);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (6, 6, 1, 'Rosu', 78);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

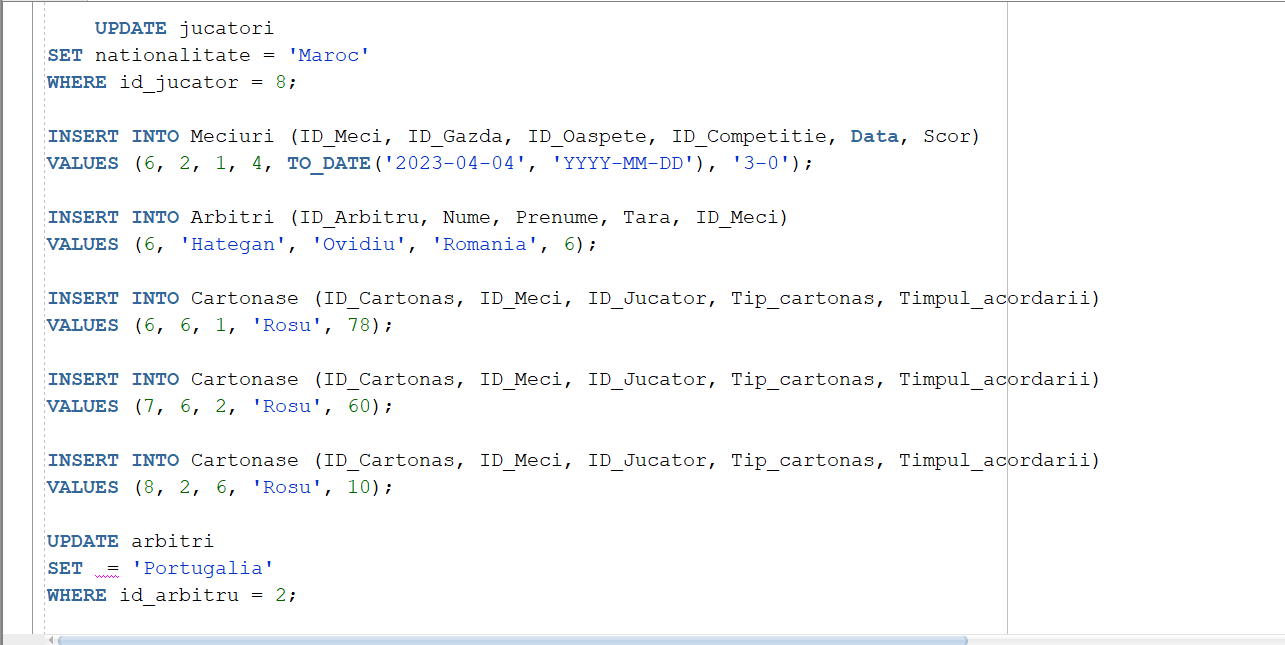
VALUES (7, 6, 2, 'Rosu', 60);

INSERT INTO Cartonase (ID\_Cartonas, ID\_Meci, ID\_Jucator, Tip\_cartonas, Timpul\_acordarii)

VALUES (8, 2, 6, 'Rosu', 10);

UPDATE arbitri

SET = 'Portugalia'

WHERE id\_arbitru = 2;  
  
****  
  
CREATE OR REPLACE PROCEDURE ex\_09(nationalitate\_param IN VARCHAR2) IS

nume jucatori.nume%type;

CURSOR c\_jucatori IS --cursor care selectează numele jucătorului

SELECT DISTINCT j.nume AS nume\_jucator

FROM jucatori j, echipe e, meciuri m, cartonase c, arbitri a

WHERE m.id\_meci = c.id\_meci and c.id\_meci = a.id\_meci and m.id\_oaspete = e.id\_echipa

and j.id\_echipa = e.id\_echipa and j.nationalitate = a.tara and j.nationalitate = nationalitate\_param and c.tip\_cartonas = 'Rosu'; --condițiile aplicate

jucator\_count NUMBER := 0;

prea\_multi EXCEPTION; --definirea de excepții

niciunul EXCEPTION;

BEGIN

OPEN c\_jucatori;

LOOP

FETCH c\_jucatori INTO nume; -- nume va lua valoarea rezultatelor

EXIT WHEN c\_jucatori%NOTFOUND;

jucator\_count := jucator\_count + 1; --un contor care să ajute la lansarea excepțiilor

END LOOP;

CLOSE c\_jucatori;

IF jucator\_count = 0 THEN --daca nu a fost gasit niciun jucator se va afisa un mesaj aferent

RAISE niciunul;

ELSIF jucator\_count > 1 THEN --asemenea in cazul in care sunt mai multi jucatori

RAISE prea\_multi;

ELSE

DBMS\_OUTPUT.PUT\_LINE (nume);

END IF;

EXCEPTION

WHEN niciunul THEN

DBMS\_OUTPUT.PUT\_LINE ('Nu exista un astfel de jucator');

WHEN prea\_multi THEN

DBMS\_OUTPUT.PUT\_LINE ('Exista mai multi jucatori');

END;

/

DECLARE

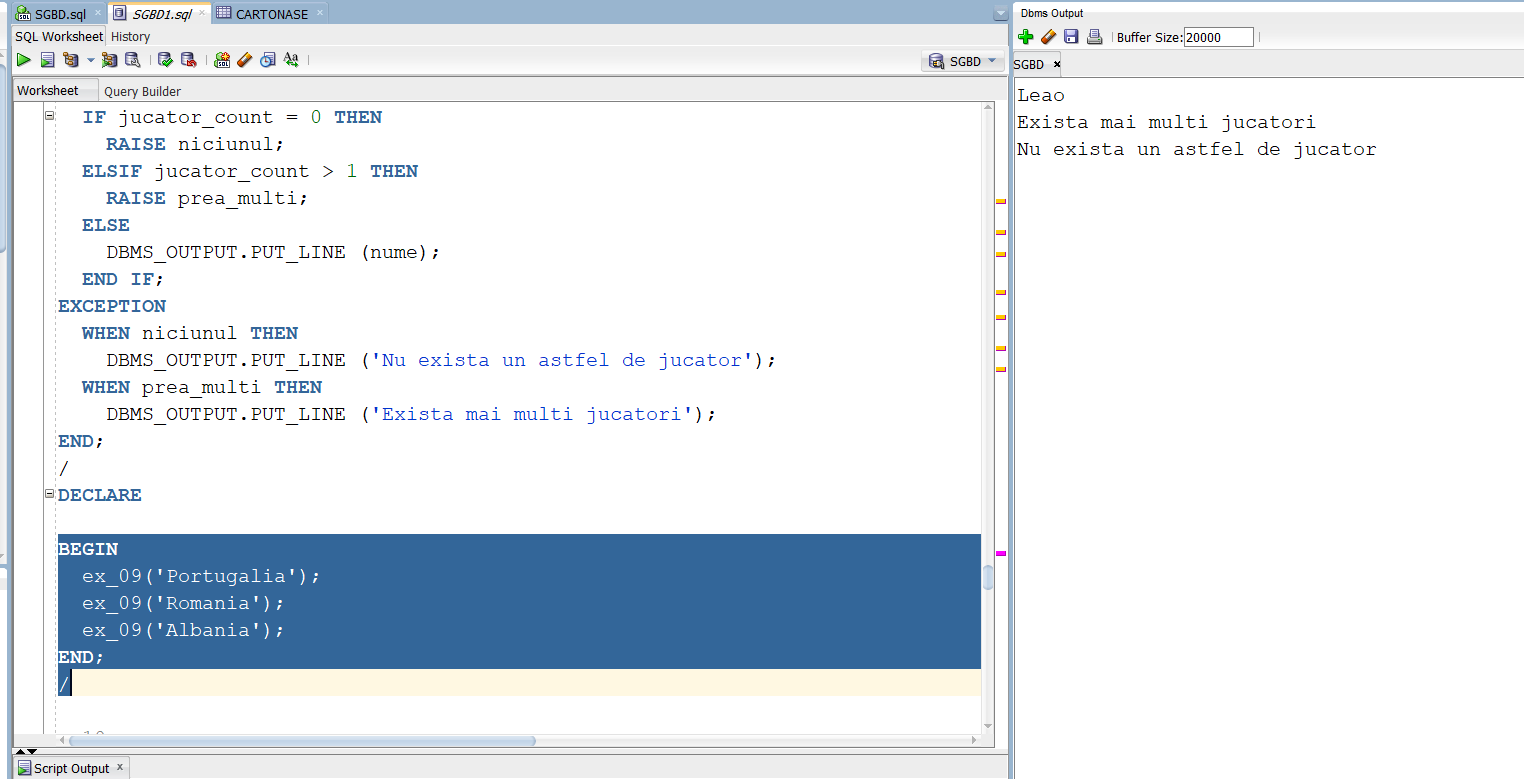
BEGIN

ex\_09('Portugalia'); --exista un singur caz, Leao

ex\_09('Romania'); --sunt doua cazuri, Chipirliu si Mihaescu

ex\_09('Albania'); -- niciun caz

END;

**  
  
10 Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.  
 --Se va defini un trigger care nu va lăsa inserarea a mai mult de 7 competiții**

CREATE OR REPLACE TRIGGER ex\_10

BEFORE INSERT OR UPDATE OR DELETE ON Competitii

DECLARE

nr NUMBER;

BEGIN

IF INSERTING OR UPDATING THEN --daca se va face o operatie de Insert sau Update

-- Verificăm dacă există înregistrări în tabelul Competitii

SELECT COUNT(\*) INTO nr FROM Competitii; --se numara totalul de competitii si se compara cu 7

IF nr = 7 THEN

RAISE\_APPLICATION\_ERROR(-20001,'Prea multe competitii');

END IF;

ELSIF DELETING THEN

-- Verificăm dacă mai există înregistrări în tabelul Competitii după ștergere

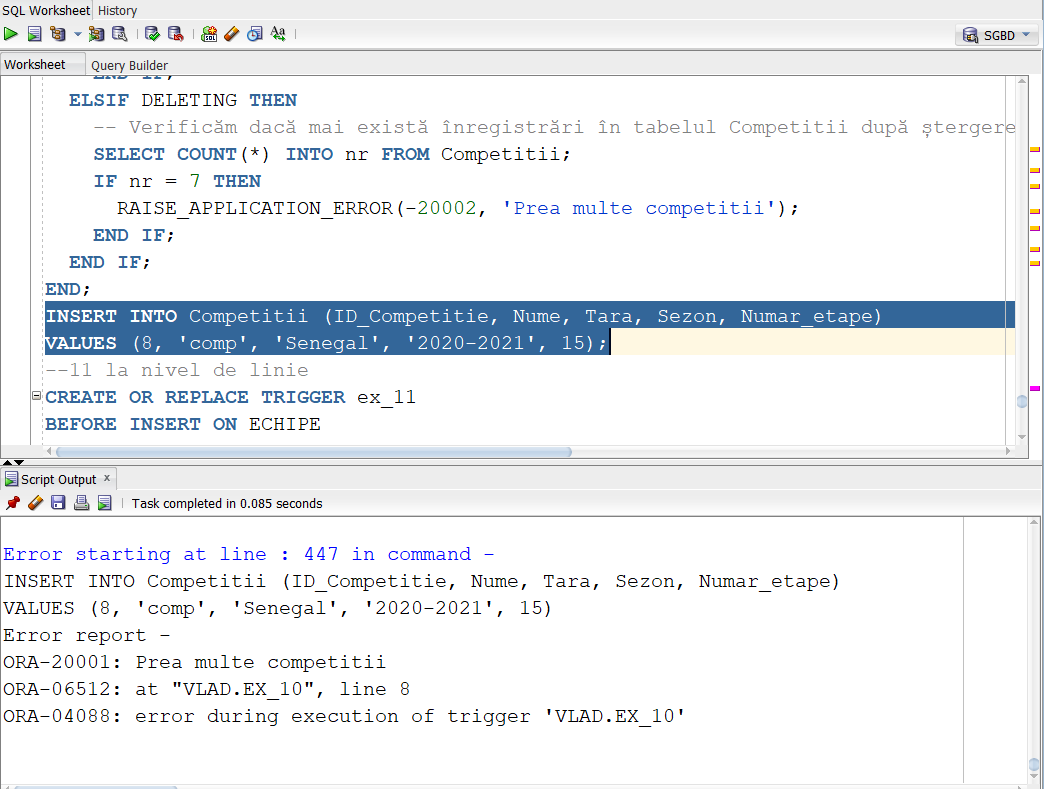
SELECT COUNT(\*) INTO nr FROM Competitii;

IF nr = 7 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Prea multe competitii');

END IF;

END IF;

END; **  
  
11** **Definiți un trigger de tip LMD la nivel de linie.Declanșați trigger-  
ul.   
  
-- un trigger care nu permitea schimbarea antrenorului cu Cristi Pustai**CREATE OR REPLACE TRIGGER ex\_11

BEFORE INSERT ON ECHIPE

FOR EACH ROW

BEGIN

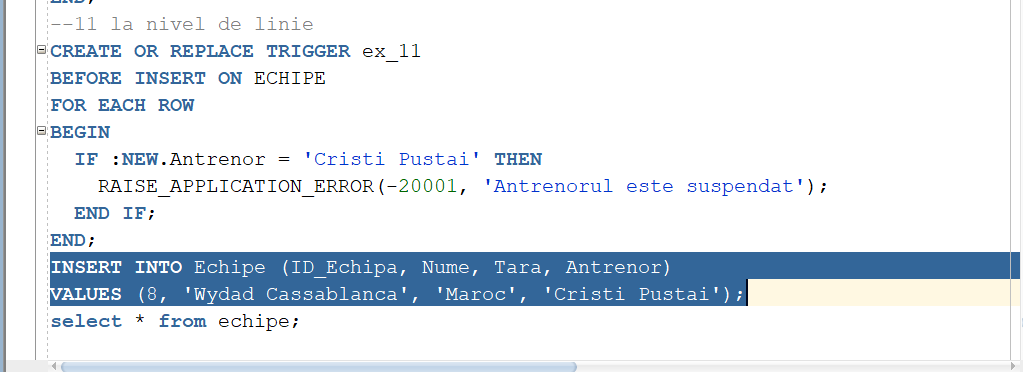
IF :NEW.Antrenor = 'Cristi Pustai' THEN --daca noul antrenor este Cristi Pustai se va lansa eroarea

RAISE\_APPLICATION\_ERROR(-20001, 'Antrenorul este suspendat');

END IF;

END;

INSERT INTO Echipe (ID\_Echipa, Nume, Tara, Antrenor)

VALUES (8, 'Wydad Cassablanca', 'Maroc', 'Cristi Pustai'); **  
12** **Definiți un trigger de tip LDD. Declanșați trigger-ul.**

**--12 Creati un trigger LDD care sa afiseze numele tabelului creat**

CREATE OR REPLACE TRIGGER ex\_12 --se creeaza tabelul ex\_12

AFTER CREATE ON SCHEMA

DECLARE

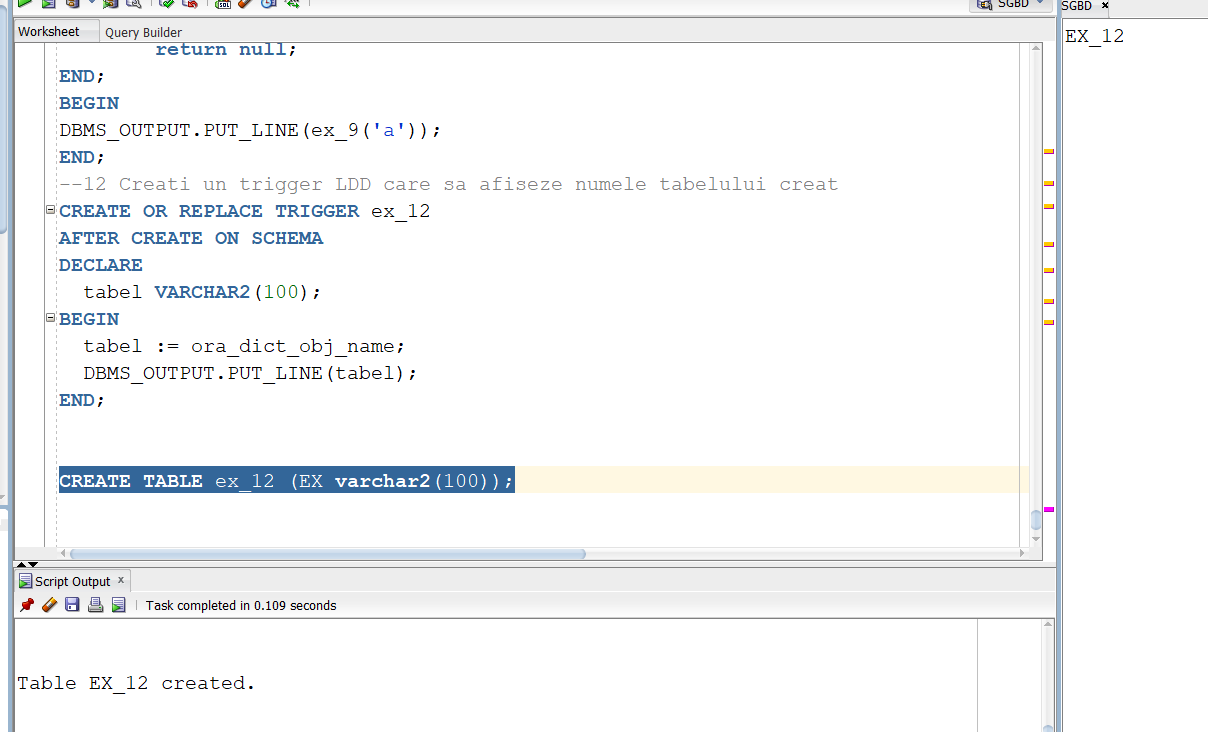
tabel VARCHAR2(100);

BEGIN

tabel := ora\_dict\_obj\_name; --tabel va lua numele noului tabel creat

DBMS\_OUTPUT.PUT\_LINE(tabel);

END;

CREATE TABLE ex\_12 (EX varchar2(100)); -- mesajul afisat la crearea tabelei  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**13.Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.**CREATE OR REPLACE PACKAGE ex\_13 AS

PROCEDURE ex\_6;

PROCEDURE ex\_7;

FUNCTION ex\_8(id\_card INTEGER) RETURN VARCHAR2;

PROCEDURE ex\_9(nationalitate\_param IN VARCHAR2);

END ex\_13;

/

CREATE OR REPLACE PACKAGE ex\_13 AS

PROCEDURE ex\_6;

PROCEDURE ex\_7;

FUNCTION ex\_8(id\_card INTEGER) RETURN VARCHAR2;

PROCEDURE ex\_9(nationalitate\_param IN VARCHAR2);

END ex\_13;

/

CREATE OR REPLACE PACKAGE BODY ex\_13 AS

PROCEDURE ex\_6 AS

TYPE jucatori IS RECORD (

nume VARCHAR2(100),

prenume VARCHAR2(100)

);

TYPE jucator IS TABLE OF jucatori INDEX BY PLS\_INTEGER;

TYPE toti\_jucatorii IS TABLE OF jucator;

players toti\_jucatorii := toti\_jucatorii();

nume\_echipa ECHIPE.nume%TYPE;

BEGIN

FOR i IN (SELECT id\_echipa FROM ECHIPE) LOOP

players.DELETE;

FOR player IN (SELECT nume, prenume FROM jucatori WHERE id\_echipa = i.id\_echipa AND data\_nasterii > TO\_DATE('1990-01-01', 'YYYY-MM-DD')) LOOP

players.EXTEND;

players(players.COUNT)(players(players.COUNT).COUNT+1) := player;

END LOOP;

SELECT nume INTO nume\_echipa

FROM ECHIPE e

WHERE i.id\_echipa = e.id\_echipa;

DBMS\_OUTPUT.PUT\_LINE(nume\_echipa);

FOR j IN 1..players.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(players(j)(1).nume);

DBMS\_OUTPUT.PUT\_LINE(players(j)(1).prenume);

END LOOP;

END LOOP;

END ex\_6;

PROCEDURE ex\_7 AS

CURSOR teams IS

SELECT id\_echipa

FROM ECHIPE;

CURSOR players (echipa INTEGER) IS

SELECT nume, prenume

FROM Jucatori

WHERE id\_echipa = echipa;

team echipe.id\_echipa%TYPE;

nJuc jucatori.nume%TYPE;

pJuc jucatori.prenume%TYPE;

team\_name echipe.nume%TYPE;

BEGIN

FOR ech IN teams LOOP

team := ech.id\_echipa;

SELECT nume INTO team\_name FROM echipe WHERE id\_echipa = team;

DBMS\_OUTPUT.PUT\_LINE(team\_name);

FOR var IN players(team) LOOP

nJuc := var.nume;

pJuc := var.prenume;

DBMS\_OUTPUT.PUT\_LINE(nJuc);

DBMS\_OUTPUT.PUT\_LINE(pJuc);

END LOOP;

END LOOP;

END ex\_7;

FUNCTION ex\_8(id\_card INTEGER) RETURN VARCHAR2 IS

exceptie\_rosu EXCEPTION;

exceptie\_prima\_repriza EXCEPTION;

nume echipe.nume%TYPE;

tip\_cartonas cartonase.tip\_cartonas%TYPE;

timpul\_acordarii cartonase.timpul\_acordarii%TYPE;

BEGIN

SELECT c.tip\_cartonas, c.timpul\_acordarii

INTO tip\_cartonas, timpul\_acordarii

FROM cartonase c

WHERE c.id\_cartonas = id\_card;

IF tip\_cartonas = 'Rosu' THEN

RAISE exceptie\_rosu;

ELSIF timpul\_acordarii < 45 THEN

RAISE exceptie\_prima\_repriza;

ELSE

SELECT e.nume

INTO nume

FROM cartonase c, meciuri m, echipe e

WHERE c.id\_cartonas = id\_card AND c.id\_meci = m.id\_meci AND m.id\_oaspete = e.id\_echipa;

RETURN nume;

END IF;

EXCEPTION

WHEN exceptie\_rosu THEN

RETURN 'Cartonasul a fost rosu';

WHEN exceptie\_prima\_repriza THEN

RETURN 'Cartonasul a fost in prima repriza';

END ex\_8;

PROCEDURE ex\_9(nationalitate\_param IN VARCHAR2) IS

nume jucatori.nume%TYPE;

CURSOR c\_jucatori IS

SELECT DISTINCT j.nume AS nume\_jucator

FROM jucatori j, echipe e, meciuri m, cartonase c, arbitri a

WHERE m.id\_meci = c.id\_meci AND c.id\_meci = a.id\_meci AND m.id\_oaspete = e.id\_echipa

AND j.id\_echipa = e.id\_echipa AND j.nationalitate = a.tara AND j.nationalitate = nationalitate\_param AND c.tip\_cartonas = 'Rosu';

jucator\_count NUMBER := 0;

prea\_multi EXCEPTION;

niciunul EXCEPTION;

BEGIN

OPEN c\_jucatori;

LOOP

FETCH c\_jucatori INTO nume;

EXIT WHEN c\_jucatori%NOTFOUND;

jucator\_count := jucator\_count + 1;

END LOOP;

CLOSE c\_jucatori;

IF jucator\_count = 0 THEN

RAISE niciunul;

ELSIF jucator\_count > 1 THEN

RAISE prea\_multi;

ELSE

DBMS\_OUTPUT.PUT\_LINE(nume);

END IF;

EXCEPTION

WHEN niciunul THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista un astfel de jucator');

WHEN prea\_multi THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multi jucatori');

END ex\_9;

END ex\_13;

/

BEGIN

ex\_13.ex\_6();

ex\_13.ex\_7();

DBMS\_OUTPUT.PUT\_LINE(ex\_13.ex\_8(1));

DBMS\_OUTPUT.PUT\_LINE(ex\_13.ex\_8(2));

DBMS\_OUTPUT.PUT\_LINE(ex\_13.ex\_8(3));

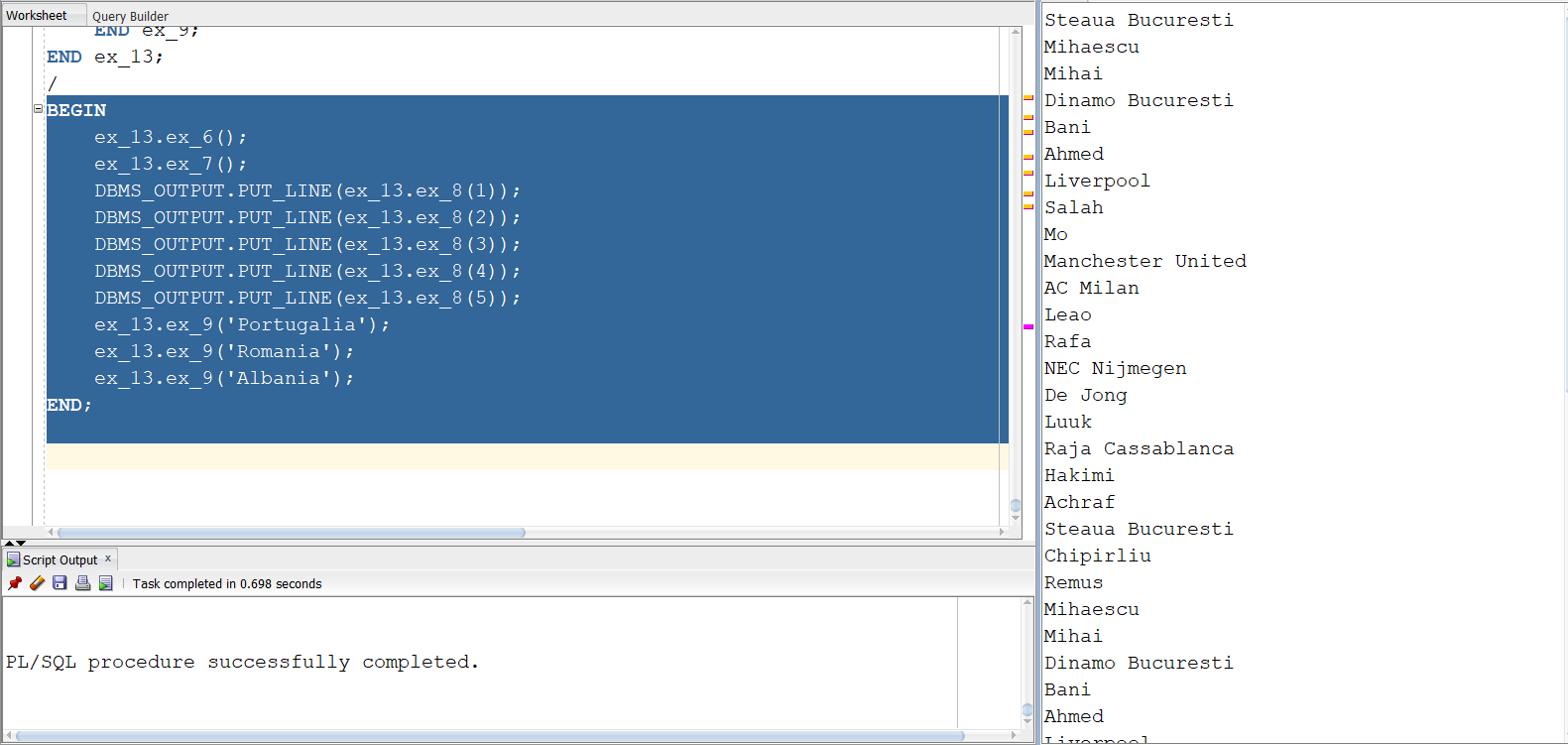
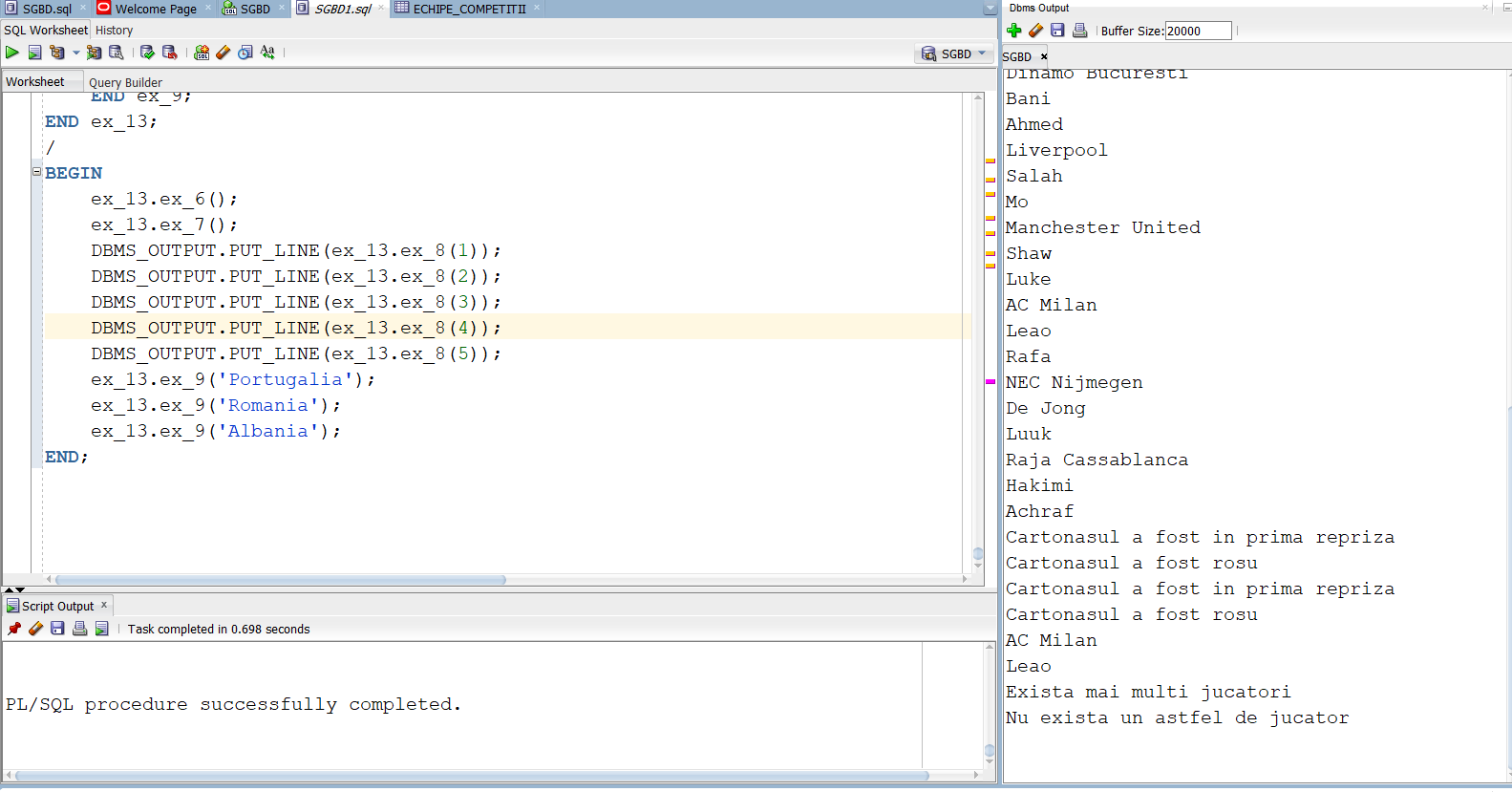
DBMS\_OUTPUT.PUT\_LINE(ex\_13.ex\_8(4));

DBMS\_OUTPUT.PUT\_LINE(ex\_13.ex\_8(5));

ex\_13.ex\_9('Portugalia');

ex\_13.ex\_9('Romania');

ex\_13.ex\_9('Albania');

END; **  
  
**